DS ASSIGNMENT NO: 06 B

Printer Spooler (Circular Queue)

Name:- Aagam Gadiya PRN :- B24CE1118
Date:-

PROBLEM STATEMENT : Printer Spooler (Circular Queue):
In a multi-user environment, printers often use a circular queue to manage print jobs. Each print jobs are added to the queue, and the printer processes them in the order they arrive. Once a print job is completed, it moves out of the queue, and the next job is processed, efficiently managing the flow of print tasks. Implement the Printer Spooler system using a circular queue without using built-in queues.

CODE

```
#include<iostream>
#include <string>
using namespace std;
class printerSpooler{
private:
string a[10];
int front, rear;
const int max = 10;
public:
printerSpooler(){
front = -1;
rear = -1;
}
void Enqueue(string val){
if ((rear + 1) % max == front){
cout << "The spooler is full!\n";
return;
}
if (front == -1 && rear == -1){
front = rear = 0;
}
else{

rear = (rear + 1) % max;
```

```
}
a[rear] = val;
cout << "Print job " << val << " added to the spooler.\n";
}
void Dequeue(){
if (front == -1 && rear == -1){
cout << "The spooler is empty!\n";
return;
}
cout << "Processing print job " << a[front] << " from the spooler.\n";
cout << "Job " << a[front] << " executed.";
if (front == rear){
front = rear = -1;
}
else{
front = (front + 1) % max;
}
}
void display(){
if (front == -1 && rear == -1){
cout << "Cannot display jobs because spooler is empty.\n";
return;
}
cout << "Current print jobs: ";
for (int i = front;; i = (i + 1) % max){
cout << a[i];
if (i == rear)
break;
cout << " , ";
}
cout << endl;
}
};
int main(){
printerSpooler q;
int choice;
string name;
do{
cout << "\n--- Printer Spooler Menu ---\n";
cout << "1. Add Print Job\n";
cout << "2. Process Print Job\n";

cout << "3. Display Print Jobs\n";
cout << "4. Exit\n";
```

```
cout << "Choose an option: ";
cin >> choice;
switch (choice){
case 1:
cout << "Enter Print Job name: ";
cin >> name;
q.Enqueue(name);
break;
case 2:
q.Dequeue();
break;
case 3:
q.display();
break;
case 4:
cout << "Exiting the program...\n";
break;
default:
cout << "Invalid option!\n";
}
}
while (choice != 4);
return 0;
}
```

OUTPUT

```
--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 1
Enter Print Job name: Job1
Print job Job1 added to the spooler.

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 1
```

```
Enter Print Job name: Job2
Print job Job2 added to the spooler.

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 3
Current print jobs: Job1 , Job2

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 2
Processing print job Job1 from the spooler.
Job Job1 executed.

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 3
Current print jobs: Job2

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 2
Processing print job Job2 from the spooler.
Job Job2 executed.

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
```

3. Display Print Jobs
4. Exit
Choose an option: 3
Cannot display jobs because spooler is empty.

--- Printer Spooler Menu ---
1. Add Print Job
2. Process Print Job
3. Display Print Jobs
4. Exit
Choose an option: 4
Exiting the program...